

Teaching structured troubleshooting: integrating a standard methodology into an information technology program

Craig Ross · R. Robert Orr

© Association for Educational Communications and Technology 2007

Abstract Troubleshooting skills are integral for the Information Technology professional. In order to address faculty concerns that students were not effectively learning required troubleshooting skills, a standardized troubleshooting methodology (the DECSAR Method) was created and integrated into the standard curriculum of a college information technology program. Components of troubleshooting were measured using a pre-/post-testing approach with the Social Problem Solving Inventory—Revised. Testing indicated improvement in several areas of troubleshooting reinforced by DECSAR. The context in which the troubleshooting methodology was applied was associated with post-test change.

Keywords Information technology · Instructional design · Teaching · Troubleshooting

The purpose of the present research was to develop a standardized troubleshooting methodology and to determine its effectiveness with a college-student population. The troubleshooting methodology is composed of six steps and will be referred to as the DECSAR Method. The six steps used in the method are: Define the problem, Examine the environment, consider the Causes, consider the Solutions, Act and test, and Review the troubleshooting. In an effort to improve the troubleshooting abilities of students, the DECSAR Method was integrated into the standard curriculum of an information technology program at a large college in Southwestern Ontario. Troubleshooting performance was assessed at two points in the semester and analysed along four dimensions that are consistent with the troubleshooting process.

C. Ross (✉) · R. R. Orr
Department of Psychology, University of Windsor, Windsor, ON, Canada N9B 3P4
e-mail: rossq@uwindsor.ca

R. R. Orr
e-mail: rrorr@uwindsor.ca

Information technology problems can be as simple as a toolbar disappearing in a word processing program, or as complicated as being unable to access certain nodes on a network. For computer users, the process of fixing problems is often referred to as troubleshooting: a process which ranges from the identification of problem symptoms to determining and implementing the action required to fix that problem (Schaafstal, Schraagen, & van Berlo, 2000).

Considering the frequency with which end users experience difficulties with technology, it should not be surprising that a large component of the training of information technology (IT) professionals is devoted to effective troubleshooting. The challenge when teaching troubleshooting skills, however, is that human beings are not inherently good troubleshooters (Teague & Allen, 1997). Partly because of a confirmatory bias, i.e., a tendency to seek out confirming information, (Teague & Allen, 1997), and partly because good troubleshooting requires intensive thinking (Gott, Lesgold, & Kane, 1997), there is a great deal that needs to be considered when developing an effective troubleshooting approach.

One of the major hurdles that must be addressed by institutions teaching troubleshooting skills is that the well-defined problems used in instructional settings are often significantly different than the ill-defined problems encountered in the real world (Jonassen, 2000). Because the problems used in instruction tend to be taken out of context and provided with artificial structure (Jonassen & Hernandez-Serrano, 2002), many traditional educational programs do not adequately prepare students for effective real-world troubleshooting (Shin, Jonassen, & McGee, 2003). Often provided with problems that have pre-determined solutions, students learn to follow a specific sequence of events to achieve that solution. Thus, instead of learning the techniques for effectively solving problems, students learning with these “structured” problems learn to follow pre-determined steps, rather than actively trying to solve the presenting problem (Roberts, 2001). As such, without the appropriate type of instruction, there can emerge a significant gap between a student’s theoretical knowledge and the student’s ability to apply knowledge in a real-life situation (Schaafstal et al., 2000).

Because troubleshooting in general is as much about making appropriate observations and tests as it is about repairing devices (Heckerman, Breese, & Rommelse, 1995), IT troubleshooters must have multiple kinds of knowledge about the IT domain. Therefore, even students provided with the best troubleshooting techniques will be unsuccessful troubleshooters if they do not possess the required domain knowledge (Shin et al., 2003). Without an appropriate understanding of the problem domain, novice troubleshooters become overwhelmed with possibilities, as they have difficulty filtering out irrelevant information (Schaafstal & Schraagen, 2000). Expert troubleshooters, however, possess a structured repository of knowledge about a domain, which allows them to determine quickly what is relevant and concentrate solely on the problem at hand (Besnard & Bastien-Toniazzo, 1999). The expert’s style of knowledge, therefore, allows for more automatic troubleshooting (Besnard & Bastien-Toniazzo, 1999) and a much more systematic approach because of the ability to represent the problem at multiple levels of abstraction (Schaafstal et al., 2000). When considered in this manner, it becomes clear that IT troubleshooting is more than just a series of decisions: it is a process requiring system knowledge, conceptual knowledge of how a system is supposed to function, an understanding of effective problem solving procedures, and an ability to manipulate effectively the problem space in order to minimize extraneous information (Jonassen, 2003).

Teaching troubleshooting

According to Teague and Allen (1997), humans bring an array of weaknesses to the troubleshooting process, including an inability to appropriately deal with uncertainty in a technical problem. Our assumption is that training can help to overcome many of these weaknesses. Gott et al. (1997) argue that effective teaching of troubleshooting has been demonstrated to help novices become more thorough and systematic. For example, according to Schaafstal et al. (2000), individuals trained in troubleshooting spent greater time trying to formulate hypotheses, while Jonassen and Hernandez-Serrano (2002) found that with training, individuals developed case-libraries of experience that could be called upon for subsequent troubleshooting. These improvements in troubleshooting performance are often the result of addressing the cognitive shortcuts and biases that humans use in day-to-day living.

One of the major factors that must be addressed in the teaching of troubleshooting is the ability to effectively gather information. Typically, human beings are quite poor at information gathering as they use heuristics, which can distort the true nature of a problem (Allen, Teague, & Carter, 1996). This type of distortion is especially prevalent in intermittent problems as it becomes difficult to determine which components of a system are actually malfunctioning (Teague & Allen, 1997).

In addition to information gathering, students must also learn to deal effectively with the mistakes and errors that are associated with the troubleshooting process. Consequently, Johnson and Chung (1999) argue it is essential that troubleshooting be taught in a realistic context.

One of the main goals in teaching troubleshooting is the development of an integrated knowledgebase of both techniques and domain information (de Croock, van Merriëboer, & Paas, 1998). Thus, it is important that students be actively involved in the process of problem formation (Dolmans, Wolfhagen, van der Vleuten, & Wijnen, 2001). One way of accomplishing this goal of involvement is to utilize a problem-based learning format (Dolmans et al., 2001). It is also important that students be gradually introduced to the troubleshooting process and have the opportunity to learn from examples of worked-out problems (Aitkinson, Renkl, & Merrill, 2003). Through a worked-out example, students can gain exposure to expert-level thinking, which may help them to more effectively organize their emerging knowledge base (Jonassen & Hernandez-Serrano, 2002).

Once students begin to develop an appreciation for the troubleshooting process, it is necessary that they begin to solve their own problems in order to understand that troubleshooting is more than just following a pre-determined series of steps. For example, there is also the requirement to utilize appropriately heuristics, which are developed through experience and supervision (Ram, Narayanan, & Cox, 1995). By becoming engaged with troubleshooting, students can also enhance their degree of domain knowledge as they become engaged with difficult problems. Although there may be concern by some that trying to solve problems and learn material at the same time will interfere with the ability to develop an effective understanding of required concepts, such is not likely the case. Previous research by deCroock et al. (1998) has indicated that delayed recall for difficult information learned under conditions of high interference is at least as good, if not better than, recall for difficult material learned without interference. As domain knowledge is one of the primary factors in effective troubleshooting (Johnson & Chung, 1999), the benefit of increased retention of information should not be overlooked: especially when it is realized that poor students are not as effective at realizing their comprehension failures as are good students (Renkl, 1997).

Considering the level of knowledge integration that is required for effective troubleshooting (Shin et al., 2003), it is important that training begin early in the educational process. By beginning training early and providing different contexts in which troubleshooting can be used, it becomes possible for students to better understand when specific strategies and techniques will be the most effective (Ram et al., 1995). Furthermore, as an understanding of the most effective techniques can be combined with an emerging understanding of the problem domain, the novice troubleshooter is able to devote more effort to the problem and spend less time trying to utilize more general problem solving approaches (Renkl & Atkinson, 2003). An early introduction to the techniques of troubleshooting, and their use in a variety of contexts, can also help the novice in transferring troubleshooting knowledge to novel situations (Schaafstal & Schraagen, 2000).

In order to encourage students to engage in the process of troubleshooting, it is important that they believe they are capable of solving the problem (Jonassen, 2000). A first step in developing these feelings of competence is to ensure that students can effectively determine the nature of the problem; without a correct problem formulation, it is nearly impossible to fix a broken system (Jonassen, 2000). This suggests that feedback is a critical component in the teaching of troubleshooting, as students must have an understanding of why their solutions failed and how they may improve in the future (Ram et al., 1995). Part of this feedback is the responsibility of the instructor, but some must also come through reflection by the troubleshooter as a process of self-diagnosis (Aitkinson et al., 2003).

General troubleshooting

Troubleshooting methods often contain a series of steps intended to identify the problem, determine the correct course of action, affect changes, and interpret the outcome (e.g., Jonassen, 2000; Schaafstal et al., 2000). This process of troubleshooting is not unique to the IT domain and analogies can be seen in other professions such as appliance repair and plumbing. One difference that may distinguish the professions, however, is that there can be a greater abstraction of knowledge required as interactions can occur between tangible components (i.e., hardware) and intangible components (i.e., software) when dealing with problems related to information technology.

Our assumption is that, regardless of the specifics of the method being used, one of the main goals of a troubleshooting methodology is to reduce the load on working memory by making the troubleshooting process more automatic. Besnard and Bastien-Toniazio (1999) argue that this reduction in memory load is the result of the expert's use of a schema-based representation, which allows for a more automatic generation of potential causes and solutions. Unfortunately, as Gott et al. (1997) note, many poor troubleshooters do not engage in the kinds of intensive skills which are required to develop a schema-based representation of knowledge. However, working memory load can also be reduced by using visual summaries to help map out the problem and troubleshooting process (McCalla, 2003).

Regardless of the process utilized, a reduction in working memory load enables the troubleshooter to devote more resources to the presenting problem and avoid becoming overwhelmed by the details of the troubleshooting process (McCalla, 2003). This reduction in working memory load is also indicative of a shift to a more rule-based approach to troubleshooting. Anderson, Fincham, and Douglass (1997) found that in the acquisition of a cognitive skill such as troubleshooting, individuals initially rely on solution examples

when solving a problem. With practice, however, novices begin to rely more on general performance rules as evidenced by the fact they no longer remember the specific examples used in instruction.

The DECSAR method

In an effort to address the troubleshooting considerations noted above, the DECSAR Method (Ross, 2004) was developed. The DECSAR Method is a six-step troubleshooting strategy that is designed to model effective troubleshooting. It was developed in conjunction with contemporary literature on troubleshooting procedure and structure and was designed to capture the fundamentals of the troubleshooting process in a way that can be easily remembered: Define the problem; Examine the situation; consider the Causes; consider the Solution; Act and test; Review the troubleshooting (please see Fig. 1).

The DECSAR Method is designed to help support a student's transition into the practice of structured troubleshooting. This is addressed by providing troubleshooting templates to help reduce working memory load and demonstrating a conceptual framework in order to enhance the student's knowledge of their own metacognitive ability (i.e., awareness of cognitive process). By following the prescribed steps in the DECSAR Method, students are encouraged to engage deliberately in the troubleshooting process. In the first step, *defining the problem*, students are instructed to carefully consider what is causing a system to malfunction in order to maximize their chances of finding an effective solution. The second step, *examine the situation*, encourages the student to make systematic observations about the system, noting both functioning and malfunctioning components and maintaining a written record of observations for future reference. The third step, *consider the causes*, requires students to propose a number of possible reasons for the malfunctioning system and then rank order them according to the likelihood that the problem is responsible for the

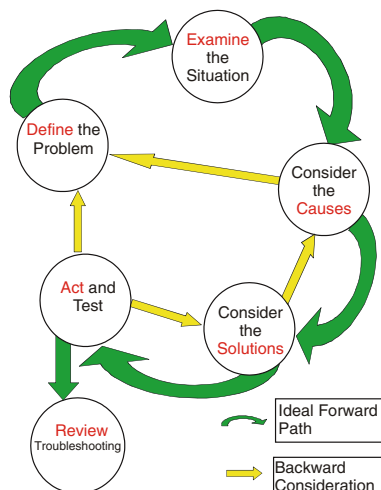


Fig. 1 The DECSAR Method. The different stages of the troubleshooting process are similar to those found in other methods. The specific contribution of the DECSAR Method is the inclusion of a more cyclical process by means of backward steps and the explicit recommendation to use the method in an interpersonal context

malfunction. The fourth step, *consider the solution*, follows the same process as that for considering causes, and ensures that a student has different courses of actions in case the first solution is ineffective. The highest-ranked solution is then used for the next step, *act and test*, where the solution is implemented and verified for efficacy. During this step, students are encouraged to compare the current functioning of the system with the notes made during the second step. Finally, once the system is repaired, students are encouraged to *review troubleshooting* in order to develop a more complete understanding of the system and to look for errors/shortcuts in their troubleshooting process.

In an effort to maximize the adoption of DECSAR by both faculty and students, it was written so as to be consistent with the methods used in information technology. However, to address concerns that students were not properly applying their knowledge, DECSAR also included an interpersonal component to encourage students to generalize their troubleshooting abilities by applying DECSAR in different contexts.

Present study

In the current study, students in an information technology program were provided with the DESCAR Method and encouraged to use it to solve problems related to software, hardware, and networking encountered during their course of study. The IT curriculum was altered so that the DECSAR Method became a component in all technical classes and was utilized by professors in classroom demonstrations. Change in troubleshooting performance was examined in a pre-/post-test fashion using a problem solving questionnaire. The purpose of this research was to validate the efficacy of the DECSAR approach based on the hypothesis that troubleshooting scores would improve over the course of the semester.

Method

Participants

Participants were drawn from the information technology program of a college located in Southwestern Ontario. Data were collected in two stages. In the first stage, participants completed the Social Problem Solving Inventory—Revised (D’Zurilla, Nezu, & Maydeu-Olivares, 2002) and a demographics page and were then introduced to the DECSAR Method. In the second stage, participants again completed the Social Problem Solving Inventory as well as a measure indicating the frequency with which they had been engaged with the DECSAR Method over the course of the term. A total of 208 students participated in the first stage of research, with 93 completing both stages.

Participants were drawn from four different areas of specialization in the information technology program. The first program ($N = 31$), office administration (Office Admin.) provided training in the use of software and management practices that are intended to support business operations and processes. The second program ($N = 18$), advanced networking (Adv. Net.) was a three-year program which focussed on the skills required to design, install, troubleshoot, and maintain a variety of operating systems and advanced network environments. The third program ($N = 24$), advanced programming (Adv. Prog.) was a three-year program, which focussed on the skills required to develop corporate and commercial software applications. The fourth program ($N = 20$) was a combination of two programming and networking streams (Prog./Net.). This last program was 2 years in

duration versus three for the other three and focussed more on the support of computer hardware and software. The final sample was composed of 46 males ($M_{\text{age}} = 21$, $SD = 4.5$), 37 females ($M_{\text{age}} = 21$, $SD = 4.0$), and 10 individuals who did not identify gender ($M_{\text{age}} = 22$, $SD = 5.1$).

Materials

For the pre-test portion of the study, each participant received a consent form, a DECSAR utilization manual, and a copy of the long form of the Social Problem Solving Inventory—Revised (D’Zurilla et al., 2002). For the post-test portion of the study, each participant received a second copy of the Social Problem Solving Inventory—Revised as well as a questionnaire which examined the frequency with which participants were exposed to the DECSAR Method (i.e., the frequency with which they saw the DECSAR Method in class) and engaged with it (i.e., the frequency with which they utilized the DECSAR Method to solve problems).

DECSAR manual

The DECSAR utilization manual (Ross, 2004) explains the importance of troubleshooting, outlines strategies for effectively learning troubleshooting, documents the DECSAR Method, demonstrates the strategies with worked-out examples, and provides a handout to help scaffold the troubleshooting process. The DECSAR handout also contains a series of practice problems that were used in class to reinforce troubleshooting skills.

Social problem solving inventory—Revised

To test the impact DECSAR had on participants’ troubleshooting skills, the Social Problem Solving Inventory—Revised (SPSI-R) was utilized. The SPSI-R is a self-report instrument that measures the ability of an individual to utilize effective solutions for the problems that are encountered in daily living. The SPSI-R is composed of 52 questions, which examine different aspects of the problem-solving process (e.g., tendency to approach problems, impulsivity in determining solutions, and rational approach to problem solving). Participants are required to endorse, on a five-point scale, the degree to which they agree with a statement which addresses a particular component of problem solving (e.g., “When I have a problem to solve, I examine what factors or circumstances in my environment might be contributing to the problem”). The SPSI-R has demonstrated good psychometric properties with an internal consistency alpha of .95 for young adults (age group of the present sample) and test–retest reliability of .87 (D’Zurilla et al., 2002). There are five subscales on this measure, but for this study only the Rational Problem Solving scale was used. The Rational Problem Solving Scale (RPS) assesses the deliberate and systematic application of problem solving strategies and techniques. This scale has some of the strongest psychometric properties in the SPSI-R with an internal consistency alpha of .92 and a test–retest reliability of .82. The RPS contains items which are most consistent with the skills encouraged by the steps of the DECSAR Method and thus taps troubleshooting ability by specifically examining the components required for effective troubleshooting performance.

The RPS contains four subscales, which correspond to four of the components of troubleshooting reinforced by the DECSAR approach. These are: (a) Problem Definition and Formation; (b) Generation of Alternative Solutions; (c) Decision Making; and

(d) Solution Implementation and Verification. The Problem Definition and Formation subscale is composed of five items which measure the degree to which a person attempts to understand a problem by gathering information. The Generation of Alternative Solutions subscale is composed of five items which measure the degree to which a person tries to generate as many alternative solutions as possible. The Decision Making subscale is composed of five items which measure the degree to which a person tries to consider the consequences of a decision. The Solution Implementation and Verification subscale is composed of five items that measure the degree to which an individual considers how his/her actions have affected the outcome of a situation.

Frequency of DECSAR exposure

This was measured using two items. The first used a 7-point scale anchored with *daily* and *never*. Participants were asked to indicate the frequency with which they had been exposed to DECSAR for all of their courses combined. The second item investigated exposure to DECSAR according to individual classes. Participants were asked to record the names of the courses for which they were enrolled and to indicate, using 5-point scales anchored with *every class* and *once per term*, the frequency with which they were exposed to DECSAR in each class.

Frequency of troubleshooting use

Using a series of four questions, participants were asked to indicate the frequency with which they used DECSAR or other troubleshooting methods to solve problems. One question utilized a 7-point scale, anchored with *daily* and *never*, to investigate the frequency with which participants used DECSAR to deal with different types of problems (i.e., school work, technical problems, interpersonal problems, and other). A second question using the same series of scales and questions was used to determine the frequency with which participants used other types of troubleshooting or problem solving methods. In order to ensure that changes seen in troubleshooting performance were attributable primarily to DECSAR, a question utilising a five-point scale anchored with *every class* and *never* asked participants to indicate the frequency with which they had been exposed to different troubleshooting or problem solving methods in class. The fourth question asked participants to rate the frequency with which they had been exposed to a troubleshooting or problem solving method other than DECSAR outside their program with a five-point scale anchored with *every day* and *never*.

Procedure

The first stage of the study occurred during the third week of the Fall semester and was conducted in class by the primary researcher as part of an initiative to introduce the DECSAR Method to all students entering into the Information Technology Program. Students were informed at the beginning of class that they were being asked to participate in the evaluation of a new troubleshooting methodology and provided with a short presentation on the meaning of informed consent. The consent forms and SPSI-R were then distributed to participants and completed *en masse*. In order to ensure that students did not feel coerced into participating, they were informed that they could fill out the SPSI with the rest of the class, but opt out of the study by writing “NO” on the top of their questionnaire.

Participants were provided with an identification number for tracking purposes and instructed to keep the identification number in an accessible place for the second stage of testing. A participant key was maintained in case identification numbers were lost.

After the questionnaire materials were collected, the primary researcher provided all students with a lecture on the DECSAR Method and distributed the DECSAR handout. At the conclusion of the formal lecture, students were placed in groups and instructed to complete one of the practice problems at the back of the handout. As these practice solutions were not part of the research study, students were graded on their participation in accordance with the usual practices of the class. Students were encouraged to use the materials supplied in the DECSAR handout to assist with their troubleshooting process and the primary researcher circulated to provide assistance as needed. Students received written feedback from the primary researcher about their troubleshooting attempts and were provided with suggestions to help improve future performance.

The second stage of research was conducted in class during the second-last week of the Fall semester. Before distributing the SPSP-R and feedback questionnaire, students were reminded of the reason for the study and the nature of informed consent. Students were also reminded that their decision to participate would have no bearing on their final grade and that they would receive a confidential feedback letter about the nature of their SPSP-R results during their final exam.

Results

Three sets of analyses were conducted. The first set of analyses utilized split-plot ANOVAs to examine the changes in troubleshooting components over the course of the semester for each of the IT programs. The second set of analyses examined the differences in use and exposure to DECSAR between each of the IT programs through a series of One-Way ANOVAs. Specifically, this analysis investigated differences in how frequently participants were exposed to DECSAR in class and how often they used DECSAR for school, technology, and personal problems. A series of planned comparisons were also included with these analyses. Due to the nature of the curriculum, it was expected that Adv. Net. and Prog./Net. would report greater exposure to DECSAR in class. Finally a series of correlations examined the factors that were associated with improvement in troubleshooting components. It was also expected that Prog./Net. would report the greatest use of DECSAR for school-based and technical problems because of the willingness to consider the method evident in the students. The final planned comparison was conducted to test the expectation that Office Admin. would demonstrate the greatest use of DECSAR for interpersonal problems because of the more interpersonal (management) focus of the program and that students in Office Admin. have relatively little involvement with the fixing of technology problems.

Changes in troubleshooting

Components of troubleshooting were drawn from the four subscales of the Rational Problem Solving scale of the SPSP-R: (a) Problem Definition and Formation (PDF); (b) Generation of Alternative Solutions (GAS); (c) Decision Making (DM); and (d) Solution Implementation and Verification (SIV). The means for each of these subscales at pre-test and post-test can be found in Table 1.

Table 1 Means and standard deviations for troubleshooting components

	PDF		GAS		DM		SIV	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Office Admin.								
Pre-test	9.23	3.92	8.24	4.07	8.16	3.28	8.00	3.57
Post-test	11.09	3.21	9.59	3.66	9.96	3.47	9.50	4.03
Adv. Prog.								
Pre-test	11.05	3.48	11.70	3.56	10.18	3.46	10.35	3.34
Post-test	11.28	3.35	11.14	3.10	9.81	3.12	9.38	4.01
Adv. Net.								
Pre-test	9.89	4.07	8.92	3.48	8.00	3.50	8.00	3.31
Post-test	9.56	3.65	10.28	3.49	8.36	3.94	8.22	3.85
Prog./Net.								
Pre-test	12.10	3.33	12.11	3.14	11.11	2.64	10.06	3.09
Post-test	12.89	2.68	12.56	3.18	11.67	3.01	10.61	3.29

Change in troubleshooting components was investigated using a series of four split-plot ANOVAs. Each analysis utilized test time (pre/post) as the within-subjects factor and IT program (four programs) as the between-subjects factor.

Problem definition and formation

The main effects for time and IT section were significant. These main effects were qualified, however, by a significant interaction between the two variables, $F(3, 84) = 3.66$, $p < .05$. A plot of means for the interaction can be found in Fig. 2. Although there were no significant differences in mean PDF scores at pre-test, a Student–Newman–Keuls analysis indicated that the PDF score for Adv. Net. was significantly lower than the PDF scores for the other sections at post-test. A series of paired sample t -tests indicated that only Office Admin. demonstrated significant change between pre-test and post-test, $t(31) = 3.87$, $p < .01$.

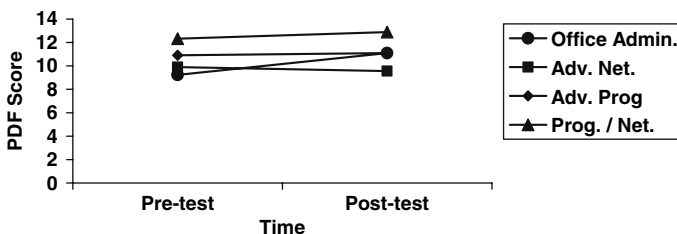
**Fig. 2** Scores for problem definition and formation

Table 2 Post-hoc results for time-collapsed score of troubleshooting components

	GAS score	DM score
Office Admin.	8.92 _a	9.06 _a
Adv. Programming	11.50 _{bc}	9.80 _{ab}
Adv. Networking	9.60 _{ab}	8.18 _a
Programming/Networking	12.33 _c	11.39 _b

Note: Means in the same column that do not share subscripts differ at $p < .05$ using the Student–Newman–Keuls post-hoc procedure

Generation of alternative solutions

A significant main effect for time was found for Generation of Alternative Solutions scores, $F(1, 84) = 4.71, p < .05$. The mean for GAS at post-test ($M = 10.93, SD = 0.37$) was significantly higher than the mean at pre-test ($M = 10.24, SD = 0.40$). There was also a main effect for IT section, $F(3, 84) = 5.59, p < .01$. Results of the Student–Newman–Keuls post-hoc test for section can be found in Table 2. Thus, the highest Generation of Alternative Solutions scores were found for Prog./Net.

Decision making

A significant main effect for section was found for Decision Making scores, $F(3, 84) = 3.92, p < .05$. Results of the Student–Newman–Keuls post hoc test for section can be found in Table 2. Thus, the highest Decision Making scores were found for Prog./Net.

Solution implementation and verification

The interaction and main effects for Solution Implementation and Verification were not statistically significant.

Difference in use and exposure

Classroom exposure to DECSAR. In order to determine how frequently participants encountered the DECSAR Method over the course of the semester, they were asked to indicate how often they had been exposed to the DECSAR Method for all of their courses combined. The One-Way ANOVA for classroom exposure was not significant. A planned contrast indicated that the average classroom exposure for the two networking programs ($M = 2.77$) was significantly more frequent than the average classroom exposure for Adv. Prog. and Office Admin. ($M = 3.6$), $t(85) = 2.4, p < .01$.

School use of DECSAR. Because the actual use of a troubleshooting methodology is as important as exposure, participants were asked to report on the frequency with which they used DECSAR for their school work. The One-Way ANOVA for school use was not significant. A planned contrast indicated that Prog./Net. used DECSAR for school problems more frequently ($M = 2.81$) than the mean for the other three programs ($M = 3.96$), $t(84) = 1.92, p < .05$.

Technical use of DECSAR. The training of the DECSAR Method encouraged participants to apply troubleshooting knowledge beyond just school-based problems. As such, participants were asked to report on the frequency with which they used DECSAR to solve technical problems. The One-Way ANOVA for technical problems was not significant. However, a planned contrast indicated that Prog./Net. used DECSAR for technical problems more frequently ($M = 3.43$) than the mean for the other three programs ($M = 4.42$), $t(83) = 2.18, p < .05$.

Interpersonal use of DECSAR. To determine if participants used DECSAR for more than just school and technical problems, participants were asked to report on the frequency with which they used DECSAR for interpersonal problems. The One-Way ANOVA for interpersonal use was significant, $F(3, 83) = 2.86, p < .05$, but a post-hoc test using the Student–Newman–Keuls procedure revealed no significant differences between the programs. A planned comparison, however, indicated that Office Admin. used DECSAR more frequently for interpersonal problems ($M = 4.23$) than the mean for the other three programs ($M = 4.87$), $t(83) = 2.04, p < .05$.

Exposure to DECSAR and types of use. Participants were asked a series of four questions to determine the frequency with which they reported being exposed to the DECSAR Method in class and the frequency with which they used DECSAR to solve different types of problems. Significant correlations were found between the frequency with which participants were exposed to the DECSAR Method in class and frequency of use for both school problems ($r = .41$) and for technical problems ($r = .36$). More frequent exposure with the method in class was associated with greater use for these two types of problems.

Uses associated with change

A series of Bonferroni-corrected correlations were conducted between each of the SPSS-R subscales and the uses of DECSAR investigated with the questionnaire distributed as part of the post-test (school use, technical use, and personal use). Two significant correlations were found, both of which were for the Solution and Implementation and Verification subscale. Greater improvement between the pre-test and post-test for SIV scores was associated with both frequency of use for technology problems ($r = .28$) and frequency of use for personal problems ($r = .27$).

On the assumption that use of DECSAR for personal problems illustrates a better generalization of its use, a series of planned comparisons were conducted which examined the change in troubleshooting scores for the Office Admin. program (most frequent interpersonal use) compared to the average change for the other three programs. Change was calculated by subtracting the pre-test score from the post-test score. The t -tests conducted indicated that the amount of change for PDF, DM, and SIV subscales for the office administration section were significantly greater than the average for the other three programs. The results from this series of analyses can be found in Table 3.

Discussion

The results of the present study indicate that the overall goal of the DECSAR Method, namely the improvement of troubleshooting performance, was met for the office administration program. This improvement is evident in the significant change for Problem Definition and Formation as well as Generation of Alternative Solutions scores on the

Table 3 Comparison of troubleshooting component differences

	Test statistic	Office Admin. difference	Mean other-program difference
PDF	$t(84) = 3.14^{**}$	1.86	0.12
GAS	$t(84) = 1.16$	1.35	0.47
DM	$t(84) = 2.30^*$	1.81	0.24
SIV	$t(85) = 2.05^*$	1.50	0.02

Note: * $p < .05$, ** $p < .01$

SPSI-R. This improvement is encouraging, considering that the initial stages of troubleshooting are often some of the most important, as they set the stage for the entire troubleshooting process (Heckerman et al., 1995).

As was expected, the networking programs, which are most involved with the troubleshooting of problems, were the ones which were most exposed to the DECSAR Method in class. This higher level of exposure is encouraging as it indicates that faculty incorporated the DECSAR method into their teaching approach. However, it is important to realize that exposure to a method does not necessarily mean that it was used effectively. One of the main reasons that the programming/networking section was analysed independently of the advanced networking program is that the advanced networking program demonstrated the most resistance to the introduction of DECSAR. Without an appropriate willingness to utilize a troubleshooting technique, it is not possible to gain the benefits of structured knowledge (de Croock et al., 1998) and reduced memory load (Jonassen, 2003) that are associated with the use of a standard troubleshooting methodology.

The importance of embracing a troubleshooting methodology can be found in the fact that the advanced networking section had pre-test scores similar to those of the office administration section, but did not demonstrate the same kind of consistent improvement, despite greater exposure to the method. One possible explanation for these findings can be found in the context of cognitive load. Because the types of problems encountered by participants in the advanced networking section were likely more complicated than those experienced by participants in office administration, the advanced networking students may not have had the cognitive resources available to attend to learning both a troubleshooting method and the required technical knowledge (Paas, Renkl, & Sweller, 2004). This may also relate to practice effects. Because the technical problems found in networking may have been more novel than those experienced in office administration, the participants in the advanced networking section may have experienced more failure and thus report less confidence with their troubleshooting abilities. Furthermore, it is likely that transfer of pre-existing skills would be more beneficial for office administration students than advanced networking students (Sweller, Chandler, Tierney, & Cooper, 1990). As Sweller and colleagues (1990) note, it is relatively common for novice problem solvers to miss out on important structural elements when trying to solve a problem. While this may not be as problematic in an interpersonal context, it certainly can be when dealing with complex computer communication protocols.

There is also an indication that the way the DECSAR method is used has a significant impact on the way in which troubleshooting scores change. The greatest improvement in scores was found for the office administration program, which is also the program that used DECSAR most frequently for interpersonal problems. With an increased use of DECSAR for interpersonal problems, the students in the office administration program were able to go beyond the technical application of the troubleshooting method. By utilizing DECSAR in a greater number of situations, students in office administration were likely able to

transfer their knowledge more effectively (Schaafstal & Schraagen, 2000). This concept of transfer may also help to explain why the networking sections, which were exposed to DECSAR more frequently in class, did not show the same kind of improvement as the office administration students. Considering that in class exposure to DECSAR was associated with higher use for both school and technical problems, it seems as though the networking students did not attempt to apply DECSAR to problems outside of those which were covered in class. By sticking to one area of use, the networking students may not have experienced the advantages of using DECSAR for other purposes. Thus, the benefits of the DECSAR Method appear to be most realized when students actually attempt to use DECSAR in their everyday problem solving endeavours. For individuals teaching problem solving to a technical group, one of the implications of this study may be to emphasize non-technical implementations of this methodology.

It is also interesting to note that office administration, the program which is traditionally the least likely to be taught a structured troubleshooting technique, is the one which seems to have benefited the most from DECSAR. This may be because participants in the office administration section had not previously attempted to troubleshoot problems. Whereas the other programs are typically comprised of individuals who have a genuine interest in computer technology and have often been involved with the technology before beginning an IT program, students in office administration tend to be more interested in learning computer-use skills and often have had significantly less interaction with computer technology. Therefore, the participants in the office administration program may have been more open to the approach proposed by DECSAR as they did not have a previous framework for troubleshooting. Likewise, those who have had a great deal of experience with computer technology may believe that they possess all of the skills needed for effective troubleshooting because of previous success with solving problems and, therefore, do not see the benefit in learning a new approach, or implementing it into a non-technical context. However, as students begin to experience more complicated problems and are more restricted by the amount of time available to solve those problems, it is possible that they would begin to better understand the benefit of using a structured troubleshooting approach such as DECSAR.

It would appear that one of the benefits of the DECSAR Method is enhancement of troubleshooting skills in non-technical areas. Therefore, one of the implications of this study is that it may be beneficial to teach troubleshooting in a broader array of contexts, even within information technology. As it appears important to increase acceptance of a method in order to increase troubleshooting skills, this factor should be looked at more extensively. This seems to be especially relevant with technically proficient individuals who may express more resistance to the learning and application of a structured model. The results of this study suggest that one of the major strengths of the DECSAR Method may be in its generalizability of use, which may result from reinforcement of the idea that recursive thinking (i.e., moving backwards through the process) is an effective component of troubleshooting.

References

- Aitkinson, R. K., Renkl, A., & Merrill, M. M. (2003). Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps. *Journal of Educational Psychology, 95*(4), 774–783.
- Allen, J. A., Teague, R. C., & Carter, R. E. (1996). The effects of network size and fault intermittency on troubleshooting performance. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 26*(1), 125–132.

- Anderson, J. R., Fincham, J. M., & Douglass, S. (1997). The role of examples and rules in the acquisition of a cognitive skill. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(4), 932–945.
- Besnard, D., & Bastien-Toniazzo, M. (1999). Expert error in trouble-shooting: An exploratory study in electronics. *International Journal of Human-Computer Studies*, 50, 391–405.
- de Croock, M. B., van Merriëboer, J. J., & Paas, F. G. (1998). High versus low contextual interference in simulation-based training of troubleshooting skills: Effects on transfer performance and invested mental effort. *Computers in Human Behavior*, 14(2), 249–267.
- Dolmans, D. H., Wolffhagen, I. H., van der Vleuten, C. P., & Wijnen, W. H. (2001). Solving problems with group work in problem-based learning: Hold on to the philosophy. *Medical Education*, 35(9), 884–889.
- D’Zurilla, T. J., Nezu, A. M., & Maydeu-Olivares, A. (2002). *Social problem-solving inventory-revised: Technical manual*. Toronto, ON: MHS.
- Gott, S. P., Lesgold, A., & Kane, R. S. (1997). Tutoring for transfer of technical competence. In S. Dijkstra, N. Seel, F. Schott, & R. D. Tennyson (Eds.), *Instructional design: International perspectives vol 2: Solving instructional design problems* (pp. 221–250). Mahwah, NJ: Lawrence Erlbaum Associates, Publishers.
- Heckerman, D., Breese, J. S., & Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3), 49–57.
- Johnson, S. D., & Chung, S. P. (1999). The effect of thinking aloud pair problem solving (TAPPS) on the troubleshooting ability of aviation technician students. *Journal of Industrial Teacher Education*, 37(1), 7–25.
- Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology, Research and Development*, 48(4), 63–85.
- Jonassen, D. H. (2003). Instructional design for learning to troubleshoot. *Performance Improvement*, 42(4), 34–38.
- Jonassen, D. H., & Hernandez-Serrano, J. (2002). Case-based reasoning and instructional design: Using stories to support problem solving. *Educational Technology: Research and Development*, 50(2), 65–77.
- McCalla, J. (2003). Problem solving with pathways. *Journal of Chemical Education*, 80(1), 92–98.
- Paas, F., Renkl, A., & Sweller, J. (2004). Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture. *Instructional Science*, 32, 1–8.
- Ram, A., Narayanan, S., & Cox, M. T. (1995). Learning to troubleshoot: Multistrategy learning of diagnostic knowledge for a real-world problem-solving task. *Cognitive Science*, 19, 289–340.
- Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science*, 21(1), 1–29.
- Renkl, A., & Atkinson, R. K. (2003). Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational Psychologist*, 38(1), 15–22.
- Roberts, L. M. (2001). Developing experimental design and troubleshooting skills in an advanced biochemistry lab. *Biochemistry and Molecular Biology Education*, 29, 10–15.
- Ross, C. (2004). *The DECSAR method: A new approach to troubleshooting* from <http://www.uwindsor.ca/users/trorrr/main.nsf>.
- Schaafstal, A., & Schraagen, J. M. (2000). Training of troubleshooting: A structured, task analytical approach. In J. M. Schraagen, S. F. Chipman, & V. L. Shalin (Eds.), *Cognitive task analysis* (pp. 57–70). Mahwah, NJ: Lawrence Erlbaum Associates Inc.
- Schaafstal, A., Schraagen, J. M., & van Berlo, M. (2000). Cognitive task analysis and innovation of training: The case of structured troubleshooting. *Human Factors*, 42(1), 75–86.
- Shin, N., Jonassen, D. H., & McGee, S. (2003). Predictors of well-structured and ill-structured problem solving in an astronomy simulation. *Journal of Research in Science Teaching*, 40(1), 6–33.
- Sweller, J., Chandler, P., Tierney, P., & Cooper, M. (1990). Cognitive load as a factor in the structuring of technical material. *Journal of Experimental Psychology: General*, 119(2), 176–192.
- Teague, R. C., & Allen, J. A. (1997). The reduction of uncertainty and troubleshooting performance. *Human Factors*, 39(2), 254–267.

Craig Ross is interested in the use of technology in the classroom and the culture of computer mediated communication among adolescents and young adults in the context of Bronfenbrenner’s ecological perspective and the general construct of social capital.

R. Robert Orr is interested in the role of technology in supporting educational interventions for young adults with learning disabilities and the implications of computer mediated communication in the social development of adolescents.